

AddPrimaryContact.apxc

```
public class AddPrimaryContact implements Queueable{

    Private Contact con;
    private String state;

    public AddPrimaryContact(Contact con, String state){
        this.con = con;
        this.state = state;
    }

    public void execute(QueueableContext context){
        List<Account> accounts = [Select Id, Name, (select FirstName, LastName, Id
from contacts)
                                from Account where BillingState = :state Limit
200];
        List<contact> primaryContacts= new List<Contact>();

        for(Account acc:accounts){
            Contact c = con.clone();
            c.AccountId = acc.Id;
            primaryContacts.add(c);
        }

        if(primaryContacts.size() >0){
            insert primaryContacts;
        }
    }

}
```

AddPrimaryContactTest.apxc

```
@isTest
public class AddPrimaryContactTest {

    static testmethod void testQueueable(){
        List<Account> testAccounts = new List<Account>();
        for(Integer i=0;i<50;i++){
            testAccounts.add(new Account(Name= 'Account '+i,BillingState='CA'));
        }
        for(Integer j=0;j<50;j++){
            testAccounts.add(new Account(Name= 'Account '+j,BillingState='NY'));
        }
        insert testAccounts;

        Contact testContact = new Contact(Firstname = 'John', LastName ='Doe');
```

```
insert testcontact;

AddPrimaryContact addit = new addPrimaryContact(testContact, 'CA');

Test.startTest();
system.enqueueJob(addit);
Test.stopTest();

    system.assertEquals(50,[Select count() from Contact where accountId
in(Select Id from Account where BillingState='CA')]);
    }
}
```